كوانتيزاسيون رنگ مبنتي بر شبكه نگاشت خود سازمانده تطبيقي با زمان پروانه‌اي

سوده كثيري بيدهندي
دانشگاه صنعتي اميركبير
kasiri@ce.aut.ac.ir

رضا صفابخش
دانشگاه صنعتي اميركبير
safa@ce.aut.ac.ir

چكيده: كوانتيزاسيون رنگ، تكنيكي براي پردازش و كاهش رنگ‌هاي تصوير است. در اين مقاله برخي از الگوريتم‌هاي كوانتيزاسيون رنگ بررسي و با يكديگر مقايسه شده‌اند. در نهايت، يك الگوريتم يادگيري مبتني بر نگاشت خودسازمانده تطبيقي با زمان پيشنهاد شده است. الگوريتم پيشنهادي از تابع نرخ يادگيري و تابع همسايگي شبكه نگاشت خودسازمانده تطبيقي با زمان، و ترتيب پرش پروانه‌اي سراسري استفاده مي‌كند. در اين روش نرخ يادگيري و تابع همسايگي جداگانه براي هر نورون درنظر گرفته مي‌شود. با داشتن شعاع همسايگي و نرخ يادگيري مستقل مي‌توان كنترل بيشتري روي رفتار هر نورون مستقل از نورون‌هاي ديگر داشت. پرش پروانه‌اي سراسري استفاده شده، ترتيب داده‌هاي ورودي را به صورت الگوهاي شبه تكراري تغيير مي‌دهد. اين ترتيب داده‌هاي آموزشي، به نورونها اجازه ميدهد تا كل ويژگي‌هاي تصوير را هر چه سريعتر درك كنند. نتايج نشان دهنده عملكرد برتر الگوريتم پيشنهادي در مقايسه با ساير الگوريتم‌هاي كوانتيزاسيون رنگ است.

واژه‌هاي كليدي: كوانتيزاسيون رنگ، شبكه نگاشت خودسازمانده، پلت رنگ

۱- مقدمه

كوانتيزاسيون رنگ، پروسه كاهش تعداد رنگهاي متمايز استفاده شده در تصوير و نمايش تصوير رنگي با استفاده از تعداد محدودي رنگ است. تصوير بدست آمده بايد از لحاظ بصري مشابه تصوير اوليه باشد. هدف اصلي كوانتيزاسيون رنگ كاهش نيازمندي رسانه ذخيره سازي و تسريع زمان ارسال تصاوير است. در يك تصوير رنگي ۲۴ بيتي *RGB*، يك بايت براي هر یک از مولفه هاي قرمز، سبز، و آبي بكار گرفته شده است. در نتيجه تعداد رنگ هاي ممكن براي هر پيكسل برابر با ۱۶/۸ ميليون رنگ است. بطور كلي يك تصوير رقمي رنگي با ۵۱۲*۵۱۲ پيكسل در فرمت غير فشرده *(24bit/pixel)* داراي حجم ۷۶۸ كيلو بايت است. بوسيله يك الگوريتم كوانتيزاسيون مناسب، و فشرده سازي با تلفات، تصوير مي تواند با كيفيت يكسان، با حجم كمتر ذخيره شود. اگر تصوير بتواند به تصويري با ۲۵۶ رنگ تبديل شود، حجم فايل تصوير به ۱/۳ حجم اوليه كاهش مي يابد. بنابراين كوانتيزاسيون رنگ نه تنها نيازمندي رسانه ذخيره سازي را كاهش مي دهد، بلكه كمك مؤثري در كاهش زمان ارسال تصاوير خواهد بود.

كوانتيزاسيون تصوير رنگي از دو مرحله اصلي، طراحي پلت رنگ مناسب براي تصوير و نگاشت هر پيكسل رنگي در تصوير به يكي از رنگهاي پلت، تشكيل شده است. طراحي پلت يعني افراز رنگهاي ورودي به N مجموعه مجزاي C_s، $(0 \le s < N)$ كه N تعداد رنگها در تصوير كوانتيزه شده

است. براي هر مجموعه رنگ C_s يك رنگ نماينده q_s وجود دارد كه در پلت رنگ قرار مي‌گيرد. بنابراين موضوع اصلي در طراحي پلت انتخاب بهترين مجموعه ممكن از رنگهاي نماينده q_s براي تصوير و افراز رنگها به N مجموعه رنگ است. در مرحله دوم هر پيكسل از تصوير، براي بدست آوردن يك تصوير با بالاترين كيفيت، به يك رنگ از q_s نگاشت مي شود. كيفيت تصويري كه ايجاد شده بطور اساسي وابسته به مرحله اول كوانتيزاسيون است. الگوريتم‌هاي كوانتيزاسيون رنگ معمولاً از يكي از دو ديدگاه الگوريتم‌هاي تجزيه كننده فضاي رنگي[1] و تجزيه‌ي خوشه اي[2] استفاده مي‌كنند. الگوريتم‌هاي تجزيه كننده فضاي رنگي، فضاي رنگي تصوير اوليه را بر اساس ضوابط مشخص به دو زيرمجموعه مجزا تجزيه مي‌كنند. سپس پروسه تجزيه بطور مكرر اعمال مي‌شود تا تعداد زيرفضاهاي مورد نظر بدست آيد. در آخر، مركز هر زيرفضا به عنوان رنگ كوانتيزه مشخص مي‌شود. الگوريتم برش ميانه(*median-cut*)[۱] محور مختصاتي كه بيشترين پراكندگي رنگي را داراست، انتخاب مي‌كند و سپس يك سطح عمود بر محور منتخب را در نظر گرفته و فضاي رنگي را به دو بخش افراز مي‌كند. در اين تقسيم تعداد پيكسل‌هاي مساوي در دو زير فضاي تقسيم شده قرار مي‌گيرد. الگوريتم برش ميانگين(*mean-cut*)[۲] مبتني بر برش ميانه[۱] است، با اين تفاوت كه فضاي رنگي را از نقطه ميانگين به جاي نقطه مياني افراز مي‌كند. الگوريتم مبتني بر واريانس[۳] مشابه برش ميانه[۱] عمل مي‌كند، با اين تفاوت كه معيار تجزيه فضاي رنگي در اين روش متفاوت است و فضاي رنگي در نقطه‌اي تقسيم مي‌شود كه واريانس در محور متناظر مينيمم شود.

چنگ(*Cheng*)[۴] الگوريتمي مبتني بر كاهش بعد فضاي رنگي پيشنهاد كرده است. اين الگوريتم در هر مرحله كلاسي كه داراي بيشترين پراكندگي رنگي است را انتخاب مي‌كند و بردارهاي رنگ آن روي يك خط تصوير مي‌شوند. اين خط بوسيله بردار ميانگين كلاس و دورترين بردار كلاس از بردار ميانگين تعريف مي‌شود. براي هر كلاس مقدار تصوير رنگهاي آن بر روي خط رسم شده محاسبه مي‌شود و بر اين اساس كلاس مورد نظر به دو زير كلاس افراز مي‌شود. در الگوريتم پيشنهادي چادها(*Chaddha*) [۵] ابتدا تصوير از یک پيش كوانتيزه كننده عبور داده مي‌شود تا تعداد رنگهاي اوليه آن كاهش داده شود. سپس كوانتيزاسيون برداري انجام مي شود تا تعداد رنگ‌هاي بدست آمده از مرحله اول به تعداد رنگ‌هاي مورد نظر كاهش يابد. بركولير(*Braquelaire*) [۶] روشهاي مبتني بر تجزيه كننده فضاي رنگي را مقايسه كرده و چند نوع بهينه سازي براي بهبود الگوريتم‌ها پيشنهاد نموده است. همچنين يك فضاي رنگي جديد $H_1H_2H_3$ پيشنهاد كرده و استدلال كرده كه كوانتيزاسيون را بهبود مي‌بخشد. در آخر، يك روش ارائه داده كه هر كلاس رنگ را در امتداد محور H_1، H_2 يا H_3 كه داراي بيشترين واريانس است افراز مي كند.

الگوريتم هاي مبتني بر تجزيه‌ي خوشه اي، فضاي رنگي را به تعداد دسته هاي مورد نظر خوشه بندي مي‌كنند. الگوريتم *k* ميانگين يكي از معروفترين الگوريتم ها در اين زمينه است. اين الگوريتم در ابتدا يك پلت رنگ اوليه را انتخاب مي‌كنند و اين پلت رنگ بطور مكرر با تغيير مراكز خوشه ها تغيير مي‌كند. كاسوگا(*Kasuga*) [۷] روشي مشابه با الگوريتم *k* ميانگين و در عين حال سريعتر ارائه كرده است. اين روش با دسته بندي مراكز *k* كلاستر در *m* دسته تعداد دفعات محاسبه فاصله ميان بردارهاي رنگ و مراكز كلاسترها را كاهش مي‌دهد. ازدمير(*Ozdemir*) [۸] يك الگوريتم فازي پيشنهاد كرده است. اين الگوريتم تعميم يافته *k* ميانگين فازي است و تابع هدف اين الگوريتم ماكسيمم كردن مقدار عضويت پيكسل‌ها به كلاسترهاي متناظر، و مينيمم كردن فاصله اقليدسي فازي بين بردار هاي رنگ تصوير اوليه و تصوير كوانتيزه شده است. تاسديزن(*Tasdizen*) [۹] مسئله كوانتيزاسيون را به وسيله الگوريتم ژنتيك مدل كرده است. در هر كروموزم، ژنها نماينده نمونه هاي رنگ تصوير هستند و مقدار هر ژن نشان مي دهد كه آن ژن عضو كدام كلاستر است. تركيبي از الگوريتم ژنتيك و الگوريتم خوشه بندي *k* ميانگين توسط شوندرز (*Scheunders*)[۱۰] پيشنهاد شده است. در اين روش هر كروموزم نماينده پلت رنگ است. پس از توليد جمعيت، الگوريتم *k* ميانگين روي هر كروموزم اعمال مي‌شود تا پلت

[1] splitting
[2] clustering

رنگ بهینه بدست آید. پي(Pei) [۱۱] الگوریتمي مبتني بر شبکه نگاشت خودسازمانده[3] ارائه کرده است. در این الگوریتم براي افزایش سرعت همگرایي شبکه، وزن اولیه نورون‌ها در امتداد قطر مکعب RGB پخش شده اند و پیکسل هاي تصویر با یك ترتیب خاص به شبکه اعمال مي‌شوند تا شبکه کل ویژگي هاي تصویر را سریعتر و بطور یکنواخت یاد بگیرد. لاي(Li)[۱۲] براساس اطلاعات لبه، یك الگوریتم کوانتیزاسیون تطبیقي ارائه داده است. دراین الگوریتم رنگهاي روي لبه به عنوان رنگهاي اصلي تصویر به صورت مکاشفه‌اي انتخاب مي‌شود. چنگ($chang$) [۱۳] یك روش یادگیري رقابتي به نام نگاشت خودسازمانده حساس به تکرار[4] ارائه کرده است. این الگوریتم با استفاده از تطبیق همسایگي شبکه نگاشت خودسازمانده و مدل یادگیري رقابتي حساس به تکرار[5] کوانتیزاسیون را انجام مي‌دهد. همچنین این نویسنده در [۱۴] الگوریتمي براي بهبود *FS-SOM* [۱۳] بر پایه کدگذاري *MSB-biased* ارائه نموده است. کرك($Kirk$)[۱۵] یك شبکه نگاشت خودسازمانده با ساختار پویا براي کوانتیزاسیون پیشنهاد داده است. در این روش رشد شبکه نگاشت خود سازمانده به وسیله‌ي میزان خطاي کوانتیزاسیون کنترل مي‌شود و پس از هر مرحله آموزش، ساختار شبکه متناسب با خطاي کوانتیزاسیون تغییر مي‌کند. البیرك($Albayrak$)[۱۶] کوانتیزاسیون رنگ را بوسیله شبکه نگاشت خودسازمانده یك بعدي و دو بعدي مدل کرده است. زیائو($Xiao$)[۱۷] روشي براي مقداردهي اولیه نورون‌ها به منظور تسریع پروسه آموزش ارائه داده است.

در این مقاله برخي الگوریتم‌هاي مبتني بر شبکه نگاشت خودسازمانده و الگوریتم چنگ[۴] که بهترین نتیجه را در مقایسه با سایر الگوریتم هاي مبتني بر تجزیه فضاي رنگي داشته است، مقایسه مي‌شوند و در نهایت یك الگوریتم تطبیقي براي کوانتیزاسیون رنگ ارائه مي‌شود. در بخش ۲ الگوریتم چنگ [۴] شرح داده شده است. در بخش هاي ۳ و۴ به ترتیب الگوریتم هاي *SOM*[۱۱] و *FS-SOM* [۱۳] بررسی مي‌شوند. در بخش ۵ الگوریتمي مبتني بر نگاشت خودسازمانده تطبیقي با زمان[6] [۱۸] ارائه مي‌شود. در بخش ٦ نتایج پیاده سازي و ارزیابي الگوریتم ها نشان داده شده و بخش آخر نتیجه گیري را در بر مي‌گیرد.

۲- الگوریتم چنگ [۴]

همان طور که در مقدمه بیان شد، این الگوریتم به دسته الگوریتم هاي مبتني بر تجزیه فضاي رنگي تعلق دارد. در این الگوریتم ابتدا رزولوشن هر مؤلفه رنگ به ۵ بیت کاهش مي یابد و یك هیستوگرام سه بعدي ۳۲*۳۲*۳۲ بر اساس تصویر رنگي ایجاد مي‌شود و بر اساس هیستوگرام بدست آمده پیکسل ها دسته بندي مي‌شود. مراحل طراحي پلت و نگاشت پیکسل ها به صورت زیر است.

۲-۱- طراحي پلت

براي دسته بندي پیکسل ها در ابتدا تمام بردارهاي رنگ به عنوان کلاستر اولیه در نظر گرفته مي‌شود، سپس کلاستري که داراي بیشترین پراکندگي رنگي(واریانس) است، تجزیه مي‌شود تا تعداد کلاستر هاي مورد نظر بدست آید.

براي هر کلاستر، بردار k که داراي بیشترین فاصله‌ي d_k از بردار میانگین $\bar{x}$ است انتخاب مي شود و خط l که از دو نقطه $\bar{x}$ و x_k عبور مي‌کند، رسم مي‌شود. مقدار تصویر[7] P_i بردار رنگ iام روي خط l، به وسیله‌ي رابطه زیر بدست مي‌آید.

$$p_i = d_k - \frac{(x_i - \bar{x}).(x_k - \bar{x})}{d_k} \qquad (۱)$$

براي تجزیه کلاسترها، میانگین تصویر بردارهاي رنگ، P_{avg} به عنوان آستانه تجزیه در نظر گرفته مي‌شود. اگر مقدار تصویر کمتر یا مساوي P_{avg} بود، بردار رنگ در کلاس اول و در غیر این صورت در کلاس دوم دسته بندي مي‌شود. کلاسها آن قدر تجزیه مي‌شوند تا تعداد کلاس هاي مورد نظر حاصل شود.

۲-۲- نگاشت پیکسل ها

مراحل نگاشت پیکسل ها به کلاس هاي بدست آمده به صورت

3 Self-Organizing Map(SOM)
4 Frequency Sensitive Self-Organizing Map(FS-SOM)
5 Frequency Sensitive Competitive Learning(FSCL)
6 Time Adaptive Self-Organizing Map(TASOM)
7 Projection

زير است.

- بردار ميانگين، $\bar{x}$ رنگهاي پلت را محاسبه كنيد.
- براي هر رنگ پلت فاصله بين آن تا بردار ميانگين را محاسبه كنيد.
- بردار رنگ با بيشترين مقدار فاصله از $\bar{x}$ را بدست آوريد، و خط l را براي تصوير كردن رسم كنيد.
- مقدار تصوير هر رنگ پلت را محاسبه، و در مجموعه S به صورت مرتب شده نگهداري كنيد.
- براي هر پيكسل x مراحل زير انجام دهيد:
  - مقدارتصوير x بر خط l، PS_x را محاسبه كنيد.
  - مقدار آستانه ε را برابر با ماكسيمم مقدار صحيح قرار دهيد.
  - يك كپي از مجموعه S در مجموعه جديد SS قرار دهيد. مراحل زير را براي پيدا كردن رنگ مناسب تكرار كنيد.
    - از مجموعه SS رنگ پلت p كه داراي نزديك ترين مقدار تصوير به PS_x است، را بوسيله جستجوي دودويي بدست آوريد.
    - اگر $\varepsilon > |PS_p - PS_x|$ است، قرار دهيد $\varepsilon = D(x,p)$ و $SS \leftarrow SS - \{p\}$.
    - در غير اين صورت از حلقه خارج شويد.
  - رنگ پيكسل را با آخرين رنگ پلت بدست آمده جايگزين كنيد.

۳- شبكه نگاشت خودسازمانده(SOM)

براي بدست آوردن پلت رنگ تصاوير رنگي از شبكه نگاشت خودسازمانده استفاده مي‌شود. بوسيله تغذيه شبكه با يك ترتيب مناسب از داده هاي ورودي، شبكه بطور خودكار عمل خوشه‌بندي داده‌ها را انجام مي‌دهد. مراحل اصلي الگوريتم عبارتند از:

- مقدار دهي اوليه وزن نورون‌ها

فرض كنيد N تعداد نورون‌هاي شبكه و بردارهاي وزن نورون‌ها، μ_i ($i=1,...,N$) نماينده بردارهاي كلاسترها هستند. در صورتي كه وزن هاي اوليه نورون‌ها به توزيع رنگ تصوير نزديك باشد، پروسه آموزش سريعتر و ساده تر خواهد بود. از آنجايي كه توزيع رنگ تصاوير طبيعي در راستاي قطر اصلي مكعب RGB است، مقادير اوليه نورون‌ها بطور يكنواخت در راستاي قطر اصلي مكعب RGB قرار مي‌گيرد. براي شبكه يك بعدي و فضاي رنگي RGB بردار هاي وزن نورون‌ها از رابطه زير بدست مي‌آيد.

$$w_i = \left[(i-1)\times\frac{255}{N},...,(i-1)\times\frac{255}{N} \right]^T \quad \forall i = 1,...,N \tag{۲}$$

- استخراج بردارهاي آموزشي از تصوير مورد نظر بر اساس ترتيب پرش پروانه اي[۸]

تصوير به $b_x \times b_y$ بلوك تقسيم مي‌شود(b_x و b_y تعداد بلوك ها در مختصات افقي و عمودي هستند). هر بلوك تصوير $blk_x \times blk_y$ پيكسل را در برمي‌گيرد و پيكسل‌ها در بلوك مربعي بطور خاصي شماره گذاري شده‌اند. در هر روبش[۹] آموزش پيكسل ها با شماره يكسان از هر بلوك به عنوان داده‌هاي آموزشي آن روبش انتخاب مي‌شوند. در تصاوير پيكسل هاي يك بلوك ۸*۸ مرتبط با يكديگرند و پراكندگي رنگ در يك بلوك تقريباً صفر است. بنابراين در هر روبش كل تصوير به شبكه آموزش داده مي‌شود. قابل توجه است در هر روبش، تابع همسايگي و تابع بهره ثابت است تا $b_x \times b_y$ پيكسل نمونه به صورت يكسان آموزش داده شوند. اين ترتيب داده هاي آموزشي، به نورون‌ها اجازه مي‌دهد تا كل ويژگي‌هاي تصوير را هر چه سريعتر درك كنند.

- پيدا كردن شبيه ترين نورون به بردار ورودي

در صورتي كه $x=[r_k, g_k, b_k]^T$ پيكسل ورودي باشد بطوريكه r_k، g_k، و b_k مولفه هاي قرمز، سبز، و آبي پيكسل ورودي باشند، بردار نورون برنده μ_c از معادله زير بدست مي‌آيد.

$$\|x-\mu_c\| = \min\|x-\mu_i\|, \quad i=1,...,N \tag{۳}$$

در آن $\|.\|$ تابع فاصله است

- بروزرساني نورون برنده و نورون‌هاي همسايه

---

[8] Butterfly jumping sequence
[9] sweep

نورون برنده و همسايگي‌هايش با استفاده از رابطه زير بروز مي‌شود.

$$\mu_i(n+1) = \mu_i(n) + \alpha . g_{ci}(n) . [x_i(n) - \mu_i(n)] \qquad (۴)$$

كه در آن c انديس نورون برنده، و $g_{ci}(n)$ تابع همسايگي اثر متقابل بين نورون c و i است. اين تابع همسايگي معمولاً يك تابع گاسي نزولي بر حسب مختصات بردارهاي r_c و r_i اين دو نورون است و از رابطه زير بدست مي‌آيد.

$$g_{ci} = \exp\left(-\left\|r_i - r_c\right\|^2 / \sigma^2\right) \qquad (۵)$$

كه در آن $\sigma = \sigma(m) = \sigma(0) \times k_1^m$ و $\sigma(m)$ شعاع همسايگي، $\alpha = \alpha(m) = \alpha(0) \times k_2^m$ تابع بهره اسكالر نزولي است، و m شماره‌ي روبش پروسه آموزش است. هر دو تابع $\sigma(m)$ و $\alpha(m)$ نزولي بر حسب زمان روبش هستند. عدد تكرار n برابر با $m.b_x.b_y$ است. k_1 و k_2 مقاديري بين ۰,۹۹-۰,۸ و $\alpha(0)$ كمتر از يك است.

- به مرحله۲ برو، تا زماني كه تابع همسايگي $g_{ci(n)}$ و تابع بهره $\alpha(m)$ به صفر نزديك شود، يا تغييرات وزن نورون‌ها ناچيز شود.

۴- شبكه نگاشت خودسازمانده حساس به تكرار(FS-SOM)

روش يادگيري رقابتي *FS-SOM* از تطبيق همسايگي شبكه نگاشت خودسازمانده، و مدل يادگيري حساس به تكرار استفاده مي‌كند و به اين ترتيب توانايي برنده شدن نورون‌هايي كه كمتر رقابت مي‌كنند را افزايش مي‌دهد. نرخ يادگيري *FS-SOM* متناسب با تعداد دفعات برنده شدن نورون‌ها است. مراحل اصلي پروسه آموزش شبكه *FS-SOM* براي طراحي پلت بقرار ذيل است:

- مقدار دهي اوليه *FS-SOM*

مقادير اوليه وزن نورون‌ها بطور يكنواخت در راستاي قطر اصلي فضاي رنگي داده‌هاي ورودي قرار مي‌گيرد، يعني

$$w_i = \left[\left(i-\frac{1}{2}\right)\times\frac{255}{N},\ldots,\left(i-\frac{1}{2}\right)\times\frac{255}{N}\right]^T \qquad (۶)$$
$$\forall i = 1,\ldots,N$$

و در آن N تعداد نورون ها است.

- جايگشت پروانه‌اي براي آرايش تصادفي ورودي

در اين الگوريتم از يك ترتيب جديد پرش پروانه‌اي سراسري براي تغيير در ترتيب ورود داده‌هاي استفاده مي‌شود كه در مقايسه با ترتيب الگوريتم *SOM* عملكرد بهتري دارد. اين ترتيب باعث مي‌شود شبكه *FS-SOM* ويژگي هاي تصوير آموزشي را سريعتر ياد بگيرد. پرش پروانه‌اي سراسري تعميم يافته پرش پروانه‌اي [۴] است. اين پرش به صورت نگاشت $\pi : Z \to Z^n$ ترتيب داده هاي ورودي $r \in Z$ به سيستم مختصات n بعدي، $(x_1, x_2, \ldots, x_n) \in Z^n$ تعريف مي شود كه در آن Z فضاي صحيح متناهي است. براي $x_i \in [0, N)$ جايگشت پروانه‌اي π از رابطه زير محاسبه مي‌شود.

$$\pi(r) = \left\{(x_1, x_2, \ldots, x_n) \in Z^n \mid x_j = \begin{cases} \sum_{i=0}^{\lceil \lg N \rceil - 1} 2^{\lceil \lg N \rceil - 1 - i}\left(r_{ni} \oplus r_{ni+j}\right) & j \neq n \\ \sum_{i=0}^{\lceil \lg N \rceil - 1} 2^{\lceil \lg N \rceil - 1 - i} r_{ni} & j = n \end{cases}\right\} \qquad (۷)$$

در اين رابطه r_i بيت iام مقدار دودويي عدد r است.

- جستجوي نورون برنده

نوروني كه كمترين فاصله را با داده ورودي دارد به عنوان نورون برنده انتخاب مي‌شود. فاصله بين نورون i و داده ورودي $x(t)$ از رابطه زير بدست مي‌آيد.

$$D_i = \left\|x(t) - w_i(t)\right\| \qquad (۸)$$

- بروزرساني نورون برنده

بردار وزن ، شمارنده بروز رساني، نورون برنده c، و همسايگي آن طبق رابطه زير بروز مي شوند.

$$w_i(n+1) = w_i(n) + G(u_c)H(i,c,m)[x(n) - w_i(n)] \qquad (۹)$$

$$u_i(n+1) = u_i(n) + H(i,c,m) \qquad (۱۰)$$

كه در آن $G(u_c)$ نرخ يادگيري حساس به تكرار، و يك تابع نزولي يكنواخت u_c است. مقدار اين تابع بين (0,1) است. $H(i,c,m)$ تابع كاهنده همسايگي است، كه ميزان بروز رساني هر نورون i، در همسايگي نورون c را در فاصله زماني گسسته، m تغيير مي‌دهد. تابع كاهنده همسايگي استفاده شده در اين الگوريتم به صورت زير است.

$$H(i,c,m) = e^{-\frac{\left\|C_i - C_c\right\|^2}{\sigma(m)^2}} \qquad (۱۱)$$

$$\sigma(m) = N\beta k^m \qquad (۱۲)$$

m شماره روبش است، و يك روبش مدت زماني است كه پارامتر هاي بروزرساني ثابت مي ماند. C_i و C_c، مختصات مكاني نورون هاي i و c است. $\sigma(m)$ اندازه همسايگي در روبش mام است و مرز تعامل بين نورون برنده c و ساير نورون ها را مشخص مي‌كند. β وسعت همسايگي اوليه را مشخص مي‌كند، و معمولاً بين 0 تا 0.5 است. مقدار ثابت $k < 1$ براي كنترل نرخ كاهش كرانه همسايگي است.

برقراري تطبيق سريع و فراموشي كند تابع يادگيري براي اطمينان از اينكه وزن‌هاي نورون‌ها در طول دوره آموزش ثابت مي‌شوند، داراي اهميت است. تابع يادگيري مورد استفاده در اين الگوريتم $G(u_c) = u_c^{-1}$ است. بدليل اينكه تابع يادگيري بطور يكنواخت كاهش نمي‌يابد و وابسته به تعداد دفعات برنده شدن نورون برنده است، نورون ها در طول پروسه آموزش بيش از حد آموزش داده نمي‌شوند.

- معيار همگرايي الگوريتم

در پايان هر روبش يادگيري، وزن نورون هاي روبش جاري و قبلي با هم مقايسه مي‌شود و در صورتي كه اين اختلاف ناچيز باشد پروسه آموزش به پايان مي‌رسد. ضابطه همگرايي به صورت زير است.

$$\frac{\sum_{i=1}^{N} \left\| w_i(m) - w_i(m-1) \right\|}{N} \le \varepsilon \qquad (۱۳)$$

براي افزايش ميزان پايداري الگوريتم، در هر فاصله آموزشي از پيش تعيين شده نورون‌هاي مرده جستجو مي‌شوند. در صورتي كه نورون هاي مرده وجود داشت، آنها دوباره مقداردهي مي‌شوند در غير اين صورت پروسه يادگيري ادامه مي‌يابد. وزن نوروني كه بطور مكرر بروز مي‌شود، روي نورون مرده كپي مي‌شود، و مقدار شمارنده بروز رساني هر دو نورون برابر با نصف مقدار شمارنده نوروني كه به طور مكرر بروز مي‌شود، قرار داده مي‌شود. در صورتي كه پارامترهاي شبكه به درستي انتخاب شود نورون‌هاي مرده وجود نخواهند داشت و در غير اينصورت نورون‌هاي مرده سريعاً پيدا و مقداردهي مي‌شوند.

۵- الگوريتم پيشنهادي

الگوريتم پيشنهادي از تابع نرخ يادگيري و تابع همسايگي شبكه نگاشت خودسازمانده تطبيقي با زمان(*TASOM*)[۱۸] و ترتيب پرش پروانه‌اي سراسري الگوريتم *FS-SOM* استفاده مي‌كند. روش يادگيري رقابتي *TASOM* از نرخ يادگيري و تابع همسايگي جداگانه براي هر نورون بهره مي‌گيرد. با داشتن شعاع همسايگي مستقل مي‌توان كنترل بيشتري روي رفتار هر نورون مستقل از نورون هاي ديگر داشت. بنابراين، به جاي استفاده از يك شعاع همسايگي مشترك براي تمام نورون‌ها در شبكه نگاشت خودسازمانده پايه بكار مي‌رود، در اين شبكه از شعاع‌هاي مستقل براي نورون‌ها استفاده مي‌شود. نرخ يادگيري و تابع همسايگي نورون ها به ازاي هر بردار ورودي بروز مي‌شود. در شبكه *TASOM* فرض بر اين است كه شبكه هيچ آگاهي از داده‌هاي ورودي ندارد، بنابراين نرخ يادگيري مستقل از زمان است و متناسب با واريانس داده‌هاي ورودي تغيير مي‌كند. مقدار واريانس به طور عددي تقريب زده مي‌شود. با توجه به ترتيب پرش پروانه‌اي الگوريتم *FS-SOM*، مي توان فرض كرد داده‌هاي ورودي به طور يكنواخت به شبكه اعمال مي‌شوند، و نرخ يادگيري نورون‌هاي شبكه را متناسب با زمان كاهش داد. بنابراين در اين روش مشابه الگوريتم *FS-SOM* داده‌هاي ورودي به ترتيب پرش پروانه‌اي سراسري به شبكه اعمال مي‌شوند و نرخ يادگيري با زمان كاهش مي‌يابد. مراحل الگوريتم به صورت زير است.

- مقدار دهي اوليه

مقادير اوليه وزن نورون‌ها بطور يكنواخت در راستاي قطر اصلي فضاي رنگي داده هاي ورودي قرار مي‌گيرد. نرخ يادگيري اوليه، $\eta_j(0)$ نزديك به يك اختيار مي‌شود. و مقادير ثابت α، β، α_s، و β_s بين صفر تا يك اختيار مي‌شوند. پارامترهاي ثابت s_f و s_g با توجه به كاربرد الگوريتم انتخاب مي‌شوند. براي بردار مقياس، $s(0) = \left[s_1(0),\ldots,s_p(0)\right]^T$ كه در آن p بعد ورودي و بردارهاي وزن است، مقادير مثبت كوچك انتخاب مي شود. نورون‌هاي همسايه هر نورون i در شبكه در مجموعه NH_i قرار مي‌گيرند. براي هر نورون i در

شـــــبكه يـــــك بعـــــدي N، $NH_i = \{i-1, i+1\}$، $NH_N = \{i-1\}$، و $NH_1 = \{2\}$ است.

- نمونه برداري

داده‌هاي ورودي طبق پرش پروانه‌اي سراسري الگوريتم *FS-SOM* به شبكه داده مي‌شوند.

- جستجوي نورون برنده

نوروني كه كمترين فاصله را با داده ورودي دارد به عنوان نورون برنده انتخاب مي‌شود. فاصله بين نورون i و داده ورودي $x(n)$ از رابطه(۱۵) بدست مي‌آيد.

$$i(x) = \arg_j \min \|x(n) - w_j(n)\|_S \quad j = 1,\dots,N \qquad (۱۴)$$

$$\|x(n) - w_j(n)\|_s = \left(\sum_k \left(\frac{(x_k(n) - w_{j,k}(n))}{s_k(n)} \right)^2 \right)^{1/2} \qquad (۱۵)$$

- بروزرساني شعاع همسايگي

شعاع همسايگي $\sigma_i(n)$ نورون برنده i به صورت زير بروز مي‌شود.

$$\sigma_i(n+1) = \sigma_i(n) + \beta \left(g\left((s_g . |NH_i|)^{-1} \times \sum_{j \in NH_i} \|w_i(n) - w_j(n)\|_s \right) - \sigma_i(n) \right) \qquad (۱۶)$$

β مقدار ثابت بين صفر و يك است. $g(\cdot)$، يك تابع اسكالر است بطوريكه براي $z > 0$، $dg(z)/dz \geq 0$ و براي نرمالسازي فاصله وزن ها استفاده مي‌شود. براي شبكه يك بعدي با N نورون، $g(0) = 0$ و $0 \leq g(z) < N$ است.

- بروزرساني پارامترهاي نرخ يادگيري

نرخ يادگيري $\eta_j(n)$ نورون‌هاي شبكه به صورت زير بروز مي‌شود.

$$\eta_j(n+1) = \eta_j(n) + \alpha \left(f\left(\frac{\|x(n) - w_j(n)\|_s}{s_f} \right) - \eta_j(n) \right) \qquad (۱۷)$$

تابع $f(\cdot)$، يك تابع اسكالر افزايشي است و براي مقادير مثبت z، $0 < f(z) \leq 1$ و $f(0) = 0$ است.

- بروزرساني وزن‌ها

بردارهاي وزن نورون‌هاي خروجي با استفاده از رابطه(۱۸) بروز مي شوند.

$$w_j(n+1) = w_j(n) + \eta_j(n+1) h_{j,i}(n+1) (x(n) - w_i(n)) \qquad (۱۸)$$

- بروزرساني بردار مقياس

بردار مقياس، $s(n) = [s_1(n), \dots, s_k(n), \dots, s_p(n)]^T$ در شبكه *TASOM* بوسيله رابطه زير بروز مي شود:

$$s_k(n+1) = \sqrt{\left(E2_k(n+1) - E_k(n+1)^2\right)^+} \qquad (۱۹)$$

و $E2_k(n+1) = E2_k(n) + \alpha_s \left(x_k^2(n) - E2_k(n) \right)$، و $E_k(n+1) = E_k(n) + \beta_s \left(x_k(n) - E_k(n) \right)$، $z^+ = \max(z, 0)$ است.

اما همانطور كه گفته شد به دليل استفاده از ترتيب پرش پروانه‌اي سراسري، داده هاي ورودي به طور يكنواخت به شبكه اعمال مي‌شوند. بنابراين بردار مقياس در اين الگوريتم متغير با زمان در نظر گرفته شده و در بازه‌هاي زماني مشخص به صورت خطي با زمان افزايش مي‌يابد.

۶- نتايج پياده سازي

به منظور مقايسه الگوريتم‌ها از تصاوير رنگي *Lena*، *Pepper*، و *Baboon* براي پياده سازي استفاده شده است. مقدار *PSNR*[10] تصوير بازسازي شده، به عنوان معيار كيفيت الگوريتم در نظر گرفته شده است. بنابراين هر چه مقدار *PSNR* بيشتر باشد، كيفيت تصوير بازسازی شده بيشتر است. مقدار *PSNR* از رابطه زير بدست مي آيد.

$$PSNR = 10 \times \log\left(\frac{3 \times 255^2}{MSE}\right) \qquad (۲۰)$$

$$MSE = \sum_{j=1}^{N_t} (X_j - \hat{X}_j) / N_t \qquad (۲۱)$$

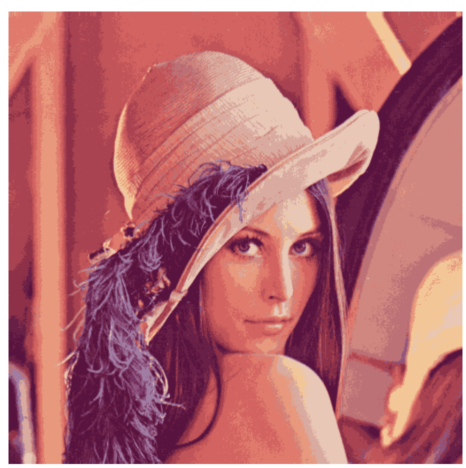
كه در آن X_j و $\hat{X}_j$ پيكسل تصوير اوليه و تصوير بازسازي شده هستند. N_t تعداد كل پيكسل‌هاي تصوير است. شكل۱ تصاوير *Lena* و *Pepper* را به همراه تصاوير بازسازي شده بوسيله الگوريتم‌هاي *FS-SOM* و الگوريتم پيشنهادي با تعداد ۱۶ رنگ را نشان مي‌دهد. جدول۱ مقايسه بين چهار الگوريتم مطرح شده را بر حسب *PSNR* براي چهار اندازه پلت نشان می دهد.

[10] Peak Signal to Noise Ratio

تصویر اولیه Lena

تصویر اولیه Pepper

تصاویر بازسازي شده بوسیله الگوریتم FS-SOM



تصاویر بازسازي شده بوسیله الگوریتم پیشنهادي

شکل۱ نتایج بازسازي تصاویر Lena و Pepper با ۱۶ رنگ

جدول۱: مقایسه الگوریتم هاي مختلف بر حسب *PSNR*

اندازه پلت	تصویر	*Cheng*	*SOM*	*FS-SOM*	الگوریتم پیشنهادي
۱۶	*Lena*	۲۷٫۲۹	۲۹٫۶۱	۲۹٫۶۷	۲۹٫۶۸
	Pepper	۲۳٫۶۱	۲۶٫۷۰	۲۶٫۷۰	۲۶٫۹۳
	Baboon	۲۳٫۶۱	۲۴٫۸۵	۲۴٫۸۹	۲۴٫۹۰
۳۲	*Lena*	۳۰٫۴۳	۳۲٫۰۹	۳۲٫۱۲	۳۲٫۱۷
	Pepper	۲۵٫۳۴	۲۹٫۱۰	۲۹٫۲۲	۲۹٫۳۴
	Baboon	۲۶٫۱۳	۲۷٫۰۵	۲۷٫۱۱	۲۷٫۱۶
۶۴	*Lena*	۳۲٫۳۹	۳۴٫۲۳	۳۴٫۲۵	۳۴٫۳۳
	Pepper	۲۷٫۷۵	۳۱٫۴۰	۳۱٫۵۹	۳۱٫۵۵
	Baboon	۲۷٫۸۲	۲۹٫۰۱	۲۹٫۱۰	۲۹٫۱۸
۱۲۸	*Lena*	۳۳٫۷۱	۳۶٫۰۸	۳۶٫۱۵	۳۶٫۲۶
	Pepper	۳۰٫۳۳	۳۳٫۴۸	۳۳٫۶۵	۳۳٫۶۵
	Baboon	۲۹٫۵۶	۳۰٫۹۱	۳۱٫۰۴	۳۱٫۱۱

جدول۱ نشان مي‌دهد که الگوریتم چنگ کمترین کارایي را در مقایسه با سایر الگوریتم‌ها دارد. الگوریتم *SOM* کارایي بهتري در مقایسه با الگوریتم چنگ دارد، و در مقایسه با دو الگوریتم *FS-SOM* و پیشنهادي ضعیف تر عمل مي‌کند. در نهایت الگوریتم پیشنهادي در مقایسه با سایر الگوریتم‌ها عملکرد بهتري دارد.

۷- نتیجه گیري

در این مقاله کوانتیزاسیون رنگ بوسیله چهار الگوریتم کوانتیزاسیون چنگ[۴]، نگاشت خودسازمانده[۱۱]، نگاشت خودسازمانده حساس به تکرار[۱۳]، و الگوریتم پیشنهادي با هم مقایسه شدند. نتایج حاصل از مقایسه این الگوریتم ها بیانگر برتري الگوریتم پیشنهادي از لحاظ کارایي در بین سایر الگوریتم‌ها است.

مراجع

[1] P. Heckbert, “Color image quantization for frame buffer display,” *Proc. Of SIG-GRAPH’82*, pp. 297-307, 1982.

[2] X. Wu, L. H. Witten “fast k-means type clustering algorithm,” *Technical Reports*, Dept. Computer Science, Univ. of Calgary, Canada, 1985.

[3] S. J. Wan, P. Prusinkiewicz, S. K. M. Wong, “Variance based color image quantization for frame buffer display,” *Color Research and Application*, vol. 15, pp 52-58, 1990.

[4] S. C. Cheng, and C. K. Yang, “A fast and novel technique for color quantization using reduction of color space dimensionality,” *Pattern Recognition Letters*, vol. 22, no. 8, pp. 845–856, 2001.

[5] N. Chaddha, W. C. Tan, and T. H. Y. Meng, “Color quantization of images based on human vision perception,” *Proc. Of IEEE Conf. Acoustics, Speech, and Signal Processing*, pp. 89–92, 1994.

[6] J. P. Braquelaire and L. Brun, “Comparison and optimization of methods of color image quantization,” *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 1048-1052, 1997.

[7] H. Kasuga, H. Yamamoto, M. Okamoto, “Color quantization using the fast K-means algorithm,” *System and Computers in Japan*, pp. 1120-1128, 1999.

[8] D. Ozdemir, and L. Akarun, “A fuzzy algorithm for color quantization of images,” *Pattern Recognition*, vol. 35, no. 8, pp. 1785–1791, 2002.

[9] T. Tasdizen, L. Akarun, and C. Ersoy, “Color quantization with genetic algorithms,” *Signal Processing: Image Communication*, vol. 12, no. 1, pp. 49-57, 1998.

[10] P. Scheunders, “A genetic approach towards optimal color image quantization,” *Proc. Of IEEE Conf. Image Processing*, pp. 1031–1034, 1996.

[11] S. C. Pei and Y. S. Lo, “Color image compression and limited display using self-organizing Kohonen map,” *IEEE Trans. Circuits and System for Video Technology*, vol. 8, no. 2, pp. 191–205, 1998.

[12] X. Li, T. Yuan, N. Yu, and Y. Yuan, “Adaptive color quantization based on perceptive edge protection,” *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3165–3176, 2003.

[13] C. H. Chang, and P. Xu, “Frequency sensitive self-organizing maps and its application in color quantization,” *Proc. Of IEEE Conf. Circuits and Systems*, pp. 804–807, 2004

[14] C. H. Chang, P. Xu, and R. Xiao, “New adaptive color quantization method based on self-organizing maps,” *IEEE Trans. Neural Networks*, vol. 16, no. 1, pp.237–249, 2005.

[15] J. S. Kirk, D. J. Chang, and J. M. Zurada, “A self-organizing map with dynamic architecture for efficient color Quantization,” *Proc. Of IEEE Conf. Neural Networks*, pp. 2128–2132, 2001.

[16] S. Albayrak, “A comparison of 1D and 2D self-organizing feature map algorithm on color image quantization,” *Proc. Of IEEE Conf. Neural Information Processing*, pp. 1291–1294, 2002.

[17] R. Xiao, C. H. Chang, and T. Srikanthan, “On the initialization and training methods for Kohonen self-organizing feature maps in color image quantization,” *Proc. Of 1st IEEE Int. Workshop on Electronic Design, Test, and Applications (DELTA'02)*, pp. 321–325, 2002.

[18] H. Shah-Hosseini, R. Safabakhsh, “New time adaptive self-organizing maps,” *IEEE Trans. System*, vol. 33, no. 2, pp. 271–282, 2003.